

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平4-248675

(43)公開日 平成4年(1992)9月4日

(51)Int.Cl.⁵

G 06 F 15/60
9/06
9/45
15/60

識別記号 庁内整理番号

3 6 0 K 7922-5 L
4 4 0 U 8944-5 B
3 6 0 D 7922-5 L
9193-5 B

F I

技術表示箇所

G 06 F 9/44 3 2 2 Z

審査請求 未請求 請求項の数3(全9頁)

(21)出願番号

特願平3-14317

(22)出願日

平成3年(1991)2月5日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者

矢野 栄一

神奈川県川崎市幸区小向東芝町1 株式会
社東芝総合研究所内

(72)発明者

増田 篤司

神奈川県川崎市幸区小向東芝町1 株式会
社東芝総合研究所内

(74)代理人

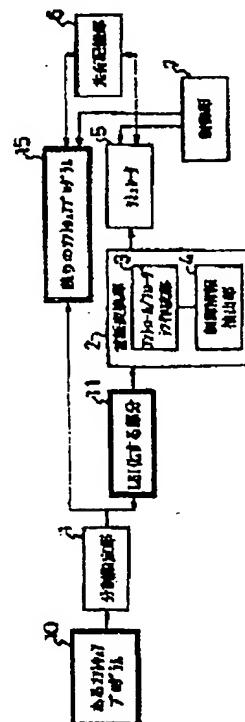
弁理士 三好 秀和 (外4名)

(54)【発明の名称】 L S I 設計支援装置

(57)【要約】 (修正有)

【目的】ソフトウェアの一部ハードウェア化を少ない作業で行なうことの出来るL S I 設計支援装置に関し、シミュレータへの入力データの設計と交換データの選定作業を軽減でき、言語変換の際に、データの流れと制御の流れを同時に表し、制御条件部分の抽出を容易に行なえるL S I 設計支援装置を提供する。

【構成】あるソフトウェアプログラムの一部をL S I 化して高速化を図るL S I 設計支援装置において、あるソフトウェアプログラム10をL S I 化する部分11と残りのソフトウェアプログラム15に分割する分割指定部1と、言語変換部2と、シミュレータ5と、共有記憶部6と、前記残りのソフトウェアプログラム15と前記シミュレータ5とを同期をとって並列実行させ前記シミュレータ5が前記共有記憶部6のデータを使用して検証し前記残りのソフトウェアプログラム15の動作を確認するよう制御する制御部7とを有して構成する。



【特許請求の範囲】

【請求項1】あるソフトウェアプログラムの一部をLSI化して高速化を図るLSI設計支援装置において、あるソフトウェアプログラムをLSI化する部分と残りのソフトウェアプログラムに分割する分割指定部と、前記LSI化する部分の記述をレジスタトランスマッピングレベルのハードウェア記述に変換する言語変換部と、前記ハードウェア記述を入力してシミュレーションを行なうシミュレータと、前記残りのソフトウェアプログラムで使用するデータと前記シミュレータで使用する回路データを保持する共有記憶部と、前記残りのソフトウェアプログラムと前記シミュレータとを同期をとって並列実行させ前記シミュレータが前記共有記憶部のデータを使用して検証し前記残りのソフトウェアプログラムの動作を確認するよう制御する制御部とを有することを特徴とするLSI設計支援装置。

【請求項2】前記言語変換部は、入力した言語記述を各種演算の演算ノード及び条件分岐等の制御ノードとそれノードを信号名と制御条件を付加したエッジで接続して構成されるコントロール/データフローグラフに変換するコントロール/データフローグラフ作成部と、作成されたコントロール/データフローグラフから暗黙の内にデータ転送に制御条件のかかっているエッジを抽出する制御情報抽出部とを有することを特徴とする請求項1に記載のLSI設計支援装置。

【請求項3】前記コントロール/データフローグラフは、制御条件の異なる同一変数名のエッジが複数本存在してもよく、ある演算ノードの1つの入力に対して複数本のエッジが存在してもよいという規則を有することを特徴とする請求項2に記載のLSI設計支援装置。

【発明の詳細な説明】

【0001】【発明の目的】

【0002】

【産業上の利用分野】本発明はLSI設計の容易化を図るLSI設計支援装置に関し、特にソフトウェアの一部ハードウェア化を少ない作業で行なうことの出来るLSI設計支援装置に関する。

【0003】

【従来の技術】近年、LSI設計の容易化を図るLSI設計支援装置において、あるソフトウェアプログラムの処理を高速化する為に、そのソフトウェアプログラムの特定の処理ブロックをLSI化し、残りの処理ブロックを制御ソフトウェアプログラムとして分担させ、元のソフトウェアプログラムの機能を実現する試みが行われている。

【0004】このようにソフトウェアプログラムで実現されている機能の一部をLSIで実現する場合に、元のソフトウェアプログラムのどの部分をLSI化するのが最も効果的であるかを考察し、また、残りの制御ソフトウェアプログラムの部分とLSI化した部分とのデータ

交換の方法をどうするかを考える必要があった。このため、元のソフトウェアプログラムの分割を変更する場合には、入力データや交換データの大幅な変更を必要とする場合があり、膨大な作業量が発生していた。

【0005】また、一般にLSIを設計する際にはシミュレータを使用して検証を行なうが、これは図9に示すように、シミュレータに対して回路を記述したテスト回路記述のファイルFI1と回路の入力信号の入力パターンが書かれている入力ファイルFI2とを与えて演算部EXでシミュレーションを実行した後、回路の出力信号の出力パターンが書かれている出力ファイルFO1を生成して、回路の論理的な検証を行なうものである。例えば、図10に示すようなテスト回路の場合、入力パターン、テスト回路記述、及び出力パターンはそれぞれ図10のようになる。このようにシミュレータに対してテスト回路毎にその入力データを作成する必要があり、あるソフトウェアプログラムの一部をLSI化する場合には、入力データの作成に際してソフトウェアプログラムの実行時の内部状態を解析して、実行の各ステップに応じてLSI部に対応する部分への入力データを与える必要があった。

【0006】具体例として、シミュレータの演算部の一部をLSI化する場合について考える。シミュレータの演算部は、本来テスト回路と同等の機能を実現するためにある所定の言語によって書かれたソフトウェアプログラムであり、その一部をLSI化することはシミュレーションの高速化に貢献する。この場合、従来では、図11に示すような全体構成で実現していた。先ず、元の演算部EXに対してテスト回路記述ファイルFI1及び入力パターンファイルFI2を入力して、シミュレーション結果である出力パターンファイルFO1を得る。次に、LSI化する部分EX1を指定して、部分EX1を取り除いた残りの演算部EX2に対するシミュレーションを行ない出力パターンFO2を得る。そしてLSI化する部分EX1(ファイルFM)を言語変換部LTでハードウェア記述ファイルFI3に言語変換して、この部分回路EX1に対する入力パターンファイルFI4を新規に作成し、それぞれの入力ファイルを入力して、部分回路EX1に対するシミュレーションを行ない出力パターンファイルFO3を得る。更に、得られた出力パターンファイルFO2及びFO3の比較を行ない対応をチェックし、得られた出力パターンファイルFO1及びFO3の対応をチェックして、出力パターンファイルFO3の内容が出力パターンファイルFO1に正しく反映されているか判断される。このように繰り返しシミュレーションを実行し、それぞれの出力ファイルを比較、対応してチェックする必要があり、その作業量は非常に大きなものであった。

【0007】また、上述の言語変換は、動作記述からレジスタトランスマッピングレベルの回路データとしてのハード

ウェア記述に変換するものであるが、従来、動作記述から内部データを作成し表示する段階では図12及び図13のように表示していた。これは制御系の流れとデータ系の流れを同時に表しているので、コントロール/データフローグラフと呼ばれているものである。この表記法においては、記述中の条件分岐の部分を表しているのがIFと書かれた分岐点で、条件が真の場合はTHENの方にデータが流れ、偽の場合はELSEの方にデータが流れることを表している。このような表記法では、条件分岐がある度にデータの流れが不明確になるという欠点があり、記述で表現されている制御情報を抽出することは出来るが、陽に表現されずに暗黙の内に制御条件がかかっているような演算に対しては、その制御条件を抽出するのは困難であった。

【0008】例えば図13では、実際には変数bの値が真の場合には $a=2$ 、偽の場合には $a=1$ として $a=a+1$ の演算が実行される記述となっているが、bの値が偽である場合に $a=1$ であることは陽には表現されておらず、同図の記述及びコントロール/データフローグラフを見ただけでこの制御情報を抽出するのは難しい。

【0009】

【発明が解決しようとする課題】以上のように、あるソフトウェアプログラムで実現されている機能の一部をLSIで実現する場合に、従来のLSI設計支援装置では、元のソフトウェアプログラムの分割を変更する際に入力データや交換データの大幅な変更を必要とする場合があり、またシミュレーションで検証する際の入力データの作成に、ソフトウェアプログラムの実行時の内部状態を解析して、実行の各ステップに応じてLSI化する部分への入力データを与える必要があり、共に膨大な作業量を発生するという欠点があった。更に、動作記述からレジスタトランスマッピングの回路データとしてのハードウェア記述に変換する言語変換において、従来の内部データの表記法では、陽に表現されずに暗黙の内に制御条件がかかっているような演算に対しては、その制御条件を抽出するのは難しいという欠点があった。

【0010】本発明は、上記問題点を解決するもので、その目的は、元のソフトウェアの分割を変更する場合にソフトウェアの内部データから自動的にシミュレータ側へのデータ設定を可能とし、LSI化する部分と残りの部分のシミュレーションの実行を同期的に行ない、内部状態を相互に交換することによりシミュレータへの入力データの設計と交換データの選定作業を軽減できるLSI設計支援装置を提供することである。また更なる目的は、動作記述からレジスタトランスマッピングの回路データへの言語変換において、データの流れと制御の流れを同時に表し、且つ制御のかかっているデータ転送のすべてが簡単に抽出でき、記述には陽に表されていない制御条件を導出できるLSI設計支援装置を提供することである。

【0011】

【課題を解決するための手段】前記課題を解決するためには、本発明の第1の特徴は、あるソフトウェアプログラムの一部をLSI化して高速化を図るLSI設計支援装置において、図1に示す如く、あるソフトウェアプログラム10をLSI化する部分11と残りのソフトウェアプログラム15に分割する分割指定部1と、前記LSI化する部分11の記述をレジスタトランスマッピングのハードウェア記述に変換する言語変換部2と、前記ハ

10 ドウェア記述を入力してシミュレーションを行なうシミュレータ5と、前記残りのソフトウェアプログラム15で使用するデータと前記シミュレータ5で使用する回路データを保持する共有記憶部6と、前記残りのソフトウェアプログラム15と前記シミュレータ5とを同期をとって並列実行させ前記シミュレータ5が前記共有記憶部6のデータを使用して検証し前記残りのソフトウェアプログラム15の動作を確認するよう制御する制御部7とを具備することである。

【0012】また、本発明の第2の特徴は、請求項1の記載のLSI設計支援装置において、図1に示す如く、前記言語変換部2は、入力した言語記述を各種演算の演算ノード及び条件分岐等の制御ノードとそれらノードを信号名と制御条件を付加したエッジで接続して構成されるコントロール/データフローグラフに変換するコントロール/データフローグラフ作成部3と、作成されたコントロール/データフローグラフから暗黙の内にデータ転送に制御条件のかかっているエッジを抽出する制御情報抽出部4とを具備することである。

【0013】また、本発明の第3の特徴は、請求項2の記載のLSI設計支援装置において、前記コントロール/データフローグラフは、制御条件の異なる同一変数名のエッジが複数本存在してもよく、ある演算ノードの1つの入力に対して複数本のエッジが存在してもよいという規則を有することである。

【0014】

【作用】本発明の第1の特徴のLSI設計支援装置では、先ず分割指定部1で、あるソフトウェアプログラム10をLSI化する部分11と残りのソフトウェアプログラム15に分割し、LSI化する部分11の記述を言語変換部2でレジスタトランスマッピングのハードウェア記述に変換し、シミュレータ5に入力として与える。次に制御部7は、残りのソフトウェアプログラム15とシミュレータ5とを同期をとって並列実行させ、シミュレータ5が残りのソフトウェアプログラム15で使用するデータとシミュレータ5で使用する回路データを保持する共有記憶部6を参照してシミュレーションを行ない、残りのソフトウェアプログラム15の動作を確認するよう制御する。

【0015】本発明の第2及び第3の特徴のLSI設計支援装置では、言語変換部2が入力した言語記述をコン

トロール／データフローグラフ作成部3で、各種演算の演算ノード及び条件分岐等の制御ノードとそれらノードを信号名と制御条件を付加したエッジで接続して構成されるコントロール／データフローグラフに変換すると、次に制御情報抽出部4で、作成されたコントロール／データフローグラフから暗黙の内にデータ転送に制御条件のかかっているエッジを抽出し、制御情報を出力する。

【0016】

【実施例】以下、本発明に係る実施例を図面に基づいて説明する。

【0017】図2に本発明の第1の実施例に係るLSI設計支援装置の全体構成図を示す。同図はあるソフトウェアプログラム10をLSI化する部分11と残りのソフトウェアプログラム15に分割して検証を行なう様子を示したものである。

【0018】先ず、あるソフトウェアプログラム10のLSI化する部分11を指定する。次に、指定された部分11を言語変換部2によりハードウェア記述に自動変換し、更にシミュレータ用の記述に変換してシミュレータ13に回路データ14として与える。また、残りのソフトウェアプログラム15には追加制御部16を、シミュレータ13の実行部には追加制御部17を追加する。そして、残りのソフトウェアプログラム15とシミュレータ13は、共有記憶領域18を共有して平行に実行され、元のソフトウェアプログラム10と同等の機能の動作を行なうか検証される。

【0019】次に、図3を用いて残りのソフトウェアプログラム15とシミュレータ13の実行時の制御関係を説明する。図3は制御関係を説明するための図であり、図3は実行のフローチャートを示す。

【0020】先ず、残りのソフトウェアプログラム15はステップS11で、シミュレータ13はステップS21で同時に起動され、シミュレータ13は回路の評価点（追加制御部17a）で待機している。他方、残りのソフトウェアプログラム15は①を実行し追加制御部16aで制御メッセージをシミュレータ13に渡し、待機する（ステップS12）。制御メッセージを受けたシミュレータ13は評価を再開して②を実行し、追加制御部17bで制御メッセージを残りのソフトウェアプログラム15に渡す（ステップS22）。残りのソフトウェアプログラム15は追加制御部16aで別処理を行なった後、追加制御部16bで待機中のところ、制御メッセージを受け取りステップ13で再評価を開始し、③を実行して終了する。

【0021】次に、本実施例に対して行なった実例を説明する。

【0022】図4は実際の元のソフトウェアプログラムに対して行なった分割とその変更の実例を示す図である。元のソフトウェアプログラム10は、表示、入力処理、特殊処理、構造の探索、及び計算の5つの機能を有

しており、先ず、実行頻度の多少に従い分割する。即ち、構造の探索及び計算のブロック43は実行頻度が多く高速処理を要する部分であることからLSI化する部分としてシミュレータ13に、また表示、入力処理、及び特殊処理のブロック41は、実行頻度が少なく高速処理を要しないことから残りのソフトウェアプログラム15に分割する。

【0023】このような初期の分割で全体のシミュレーションを行なった結果、共有記憶領域18への参照回数

10 が多すぎてスピードネックとなっていることが判明した。このため特殊機能をシミュレータ13と残りのソフトウェアプログラム15に分割して全体を再分割するよう変更を行ない、全体のシミュレーションを行なったところ、共有記憶領域18への参照回数が減少し、全体の処理速度の向上が図れた。尚、最終的に分割の再指定を数回行ない、また併せて処理構造の変更も行なった。

【0024】次に、本発明の第2の実施例に係るLSI設計支援装置を図5を用いて説明する。図5は従来技術の具体例として説明したシミュレータの演算部の一部を

20 LSI化する場合に本発明を適用したものである。同図において、図11（従来例）と重複する部分には同一の符号を附して説明を省略する。

【0025】先ず、LSI化する部分EX1を指定して、言語変換部2でハードウェア記述ファイルF13に言語変換して、シミュレータSYM2の入力とする。次に、元の演算部EXからLSI化する部分EX1を取り除いた残りの部分EX2に追加制御部EXC2を付加し、またLSI化する部分EX1に追加制御部EXC1を付加し、シミュレータSYM1に対してテスト回路記述ファイルF11及び入力パターンファイルF12を入力して、シミュレータSYM1とシミュレータSYM2を同時に平行して実行してシミュレーション結果である出力パターンファイルFO1を得て検証を行なう。この時の制御関係は第1の実施例と同様である。

【0026】このように本実施例では、従来行なっていた部分回路EX1に対する入力パターンファイルF14の新規作成、部分EX1を取り除いた残りの演算部EX2に対するシミュレーション（出力パターンFO2）、部分回路EX1に対するシミュレーション（出力パターンFO3）、得られた出力パターンファイルFO2及びFO3の比較対応チェック、及び得られた出力パターンファイルFO1及びFO3の対応チェックの各作業が不要となり、シミュレータへの入力データの設計と交換データの選定作業を軽減できる。

【0027】次に、本発明の第3の実施例を図を用いて説明する。

【0028】図6は本発明の第3の実施例に係るLSI設計支援装置の全体構成図を示す。同図において、61は制御情報を抽出しようとする動作記述、62は記述からエッジに信号名と制御情報を付加したコントロール／

データフローグラフを作成するコントロール/データフローグラフ作成部、63はこれにより出力されたコントロール/データフローグラフ、64はコントロール/データフローグラフの各エッジに対する変数名と条件を記載した制御情報テーブル、65はこのコントロール/データフローグラフを基に制御情報を抽出する制御情報抽出部、66は出力となる制御情報である。

【0029】次に、実際の具体例を用いて説明する。本発明によるコントロール/データフローグラフの表記法では、円の中に+、-、×の演算子の記されたものが演算のノードであり、矩形で表されたものが制御のノードである。そしてそれらのノード間を線分で接続しているものがエッジである。また、制御条件の異なる同一変数名のエッジが複数本存在してもよく、ある演算ノードの1つの入力に対して複数本のエッジが存在してもよいという規則を持つ。

【0030】図7は第1の具体例で、記述とそのコントロール/データフローグラフを示している。また、表1はコントロール/データフローグラフの各エッジに対して信号名と制御条件を記したものである。即ち、動作記述61をコントロール/データフローグラフ作成部62に入力し、コントロール/データフローグラフ63を生成すると共に、エッジ毎に信号名、制御されている条件を明記したテーブル64を作成する。表1ではエッジe2に制御条件がかかっているだけであるが、同一の演算の入力に接続しているe1には制御条件がかかっていない*

*い。そこで制御条件抽出部65ではこの演算ノードの1つの入力に対して複数のエッジが入力されているノードを中心に解析を行なって、e1のエッジに対してe2のエッジの条件と併的的な条件（即ち、変数yの値が真の時はa=2、偽の時はa=1）を付加し制御情報66を更新する。この更新後の制御条件を表したもののが表2である。このように本実施例によれば、従来制御条件がなかったエッジに対して制御条件を付加することが可能となり、記述中には陽に表現されていない制御条件を導出できる。

【0031】

【表1】

エッジ名	信号名	制御エッジ名
e 1	a	-
e 2	a	e 5
e 3	-	-
e 4	a	-
e 5	-	-
e 6	-	-
e 7	y	-

<表. 1>

【0032】

【表2】

エッジ名	信号名	制御エッジ名
e 1	a	e 6
e 2	a	e 5
e 3	-	-
e 4	a	-
e 5	-	-
e 6	-	-
e 7	y	-

<表. 2>

【0033】次に、図8は第2の具体例で、記述とそのコントロール/データフローグラフを示している。また、表3はコントロール/データフローグラフの各エッジに対して信号名と制御条件を記したものである。表3では制御条件がかかっているエッジはe9、e10だけである。制御条件抽出部65ではこれらのエッジを通り、エッジe1、e2、e5、e6にエッジe9と同じ条件が、またエッジe3、e4、e7、e8にエッジe10と同じ条件がかかっていることを見い出し、制御情報を更新する。この更新後の制御条件を表したもののが表4である。このように本実施例によれば、記述中には陽に表現されていない文に対する制御条件も抽出できる。

【0034】

【表3】

*【0035】

【表4】

エッジ名	信号名	制御エッジ名
e 1	b	-
e 2	c	-
e 3	b	-
e 4	c	-
e 5	X 1	-
e 6	-	-
e 7	X 2	-
e 8	-	-
e 9	a	e 13
e 10	a	e 14
e 11	-	-
e 12	a	-

<表. 3>

10

エッジ名	信号名	制御エッジ名
e 1	b	e 13
e 2	c	e 13
e 3	b	e 14
e 4	c	e 14
e 5	X 1	e 13
e 6	-	e 13
e 7	X 2	e 14
e 8	-	e 14
e 9	a	e 13
e 10	a	e 14
e 11	-	-
e 12	a	-

<表. 4>

【0036】以上の例のように、制御情報を抽出した後で回路構成を考えることで、条件によっては不必要的演算（例えば、論理的に成立し得ない条件がかかる流れにある演算等）を削除したり、排他的な条件の基にある（演算器は同時に実行されることがないことから）演算器同士を同一のハードウェアで表現したりするということが考え易くなる。

【0037】

【発明の効果】以上のように本発明によれば、ソフトウェアプログラムの一部をLSI化する場合に、LSI化する部分と残りの部分のシミュレーションの実行を同期的に行ない、内部状態を相互に交換することとしたので、検証の際のシミュレーションに使用する入力データの作成作業を軽減でき、また、LSI化する範囲を変更する場合にも自動的に制御データを付加することが可能となり交換データの選定作業を軽減できるLSI設計支

援装置を提供することができる。

【0038】また本発明によれば、動作記述からレジスタトランスマッピングの回路データへ言語変換する場合、データの流れと制御の流れを同時に表し、且つ制御条件部分の抽出を容易に行なうことができ、記述には陽に表されていない制御条件をも導出できるので、質の良い回路設計が可能なLSI設計支援装置を提供することができる。

【図面の簡単な説明】

【図1】本発明のクレーム対応図である。

【図2】本発明の第1の実施例に係るLSI設計支援装置の全体構成図である。

【図3】第1の実施例の残りのソフトウェアプログラムとシミュレータの実行時の制御関係を説明する図である。

【図4】第1の実施例において実際の元のソフトウェア

プログラムに対して行なった分割とその変更の実例を示す図である。

【図5】本発明の第2の実施例に係るLSI設計支援装置の全体構成図である。

【図6】本発明の第3の実施例に係るLSI設計支援装置の全体構成図である。

【図7】本発明の第3の実施例の第1の具体例に係る動作記述とコントロール/データフローラグラフである。

【図8】本発明の第3の実施例の第2の具体例に係る動作記述とコントロール/データフローラグラフである。

【図9】従来のシミュレータと入出力ファイルの関係を説明する図である。

【図10】従来のシミュレータの入出力ファイルの具体例である。

【図11】従来のシミュレータを使用して演算部の一部をLSI化する場合の全体構成図である。

【図12】従来の動作記述とコントロール/データフローラグラフの第1の具体例である。

【図13】従来の動作記述とコントロール/データフローラグラフの第2の具体例である。

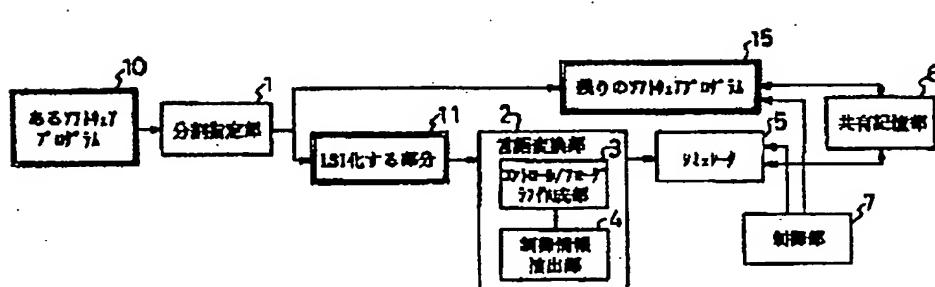
【符号の説明】

1 分割指定部

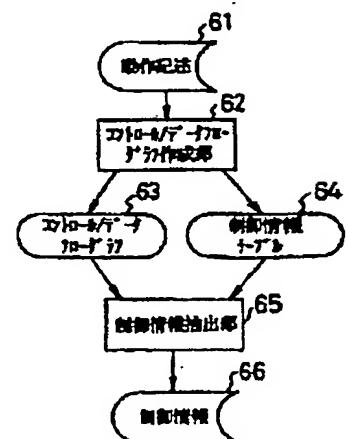
2 言語変換部

- 3 コントロール/データフローラグラフ作成部
- 4 制御情報抽出部
- 5 シミュレータ
- 6 共有記憶部
- 7 制御部
- 10 あるソフトウェアプログラム
- 11 LSI化する部分
- 12 ハードウェア記述
- 13 シミュレータ
- 10 15 残りのソフトウェアプログラム
- 16 追加制御部
- 16 a 追加制御部
- 16 b 追加制御部
- 17 追加制御部
- 17 a 追加制御部
- 17 b 追加制御部
- 18 共有記憶領域
- 61 動作記述
- 62 コントロール/データフローラグラフ作成部
- 20 63 コントロール/データフローラグラフ
- 64 制御情報テーブル
- 65 制御情報抽出部
- 66 制御情報

【図1】

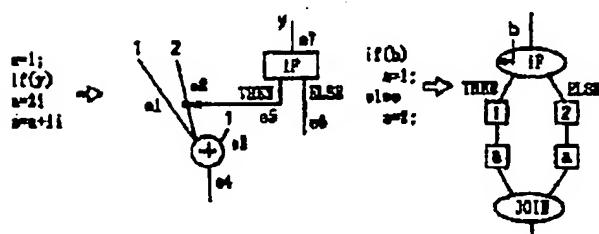


【図6】

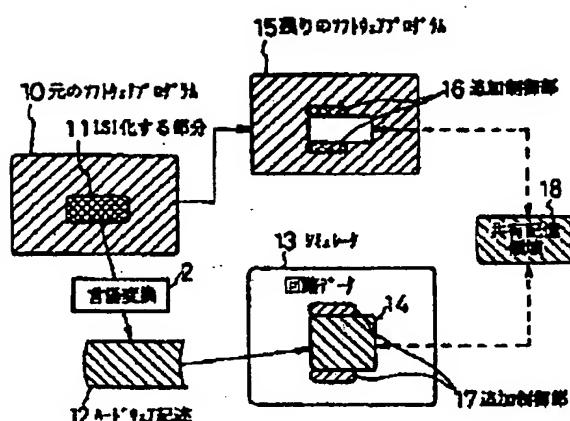


【図7】

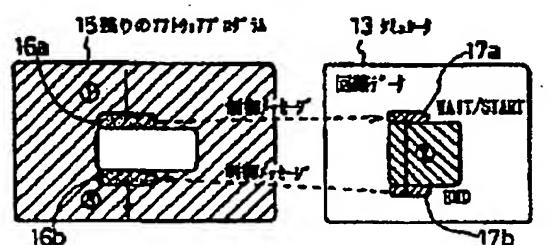
【図12】



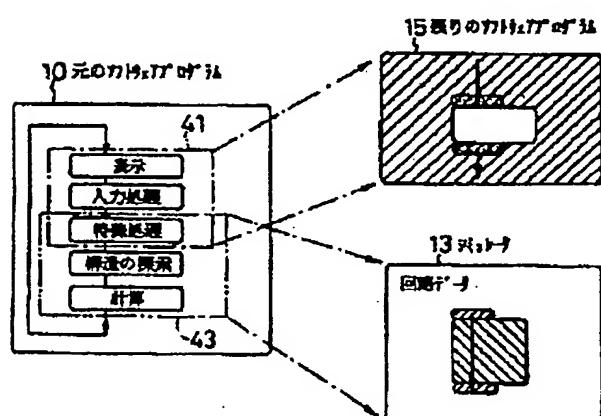
【図2】



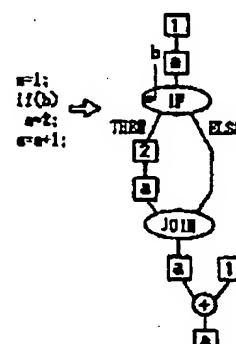
【図3】



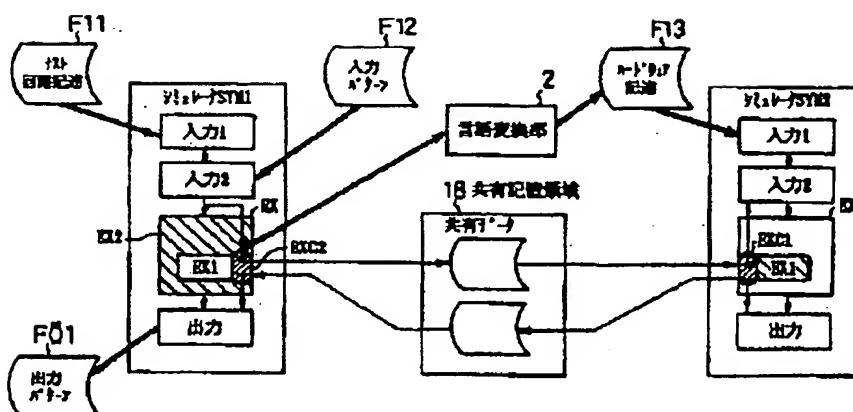
【図4】



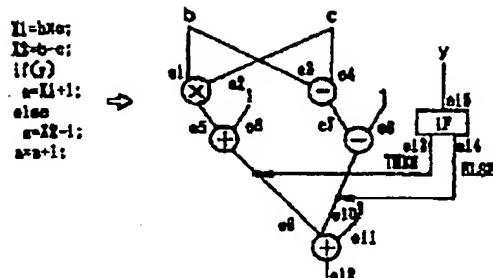
【図13】



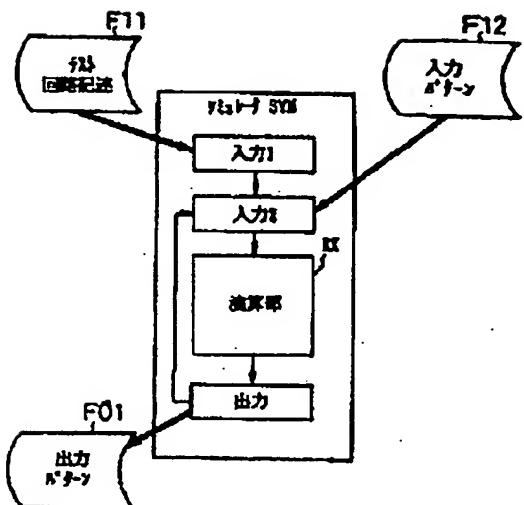
【図5】



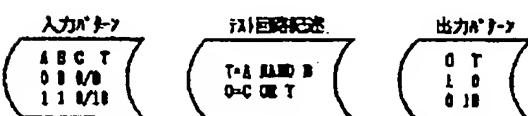
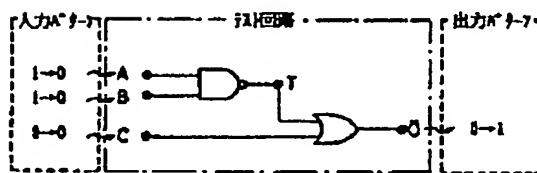
【図8】



【図9】



【図10】



【図11】

